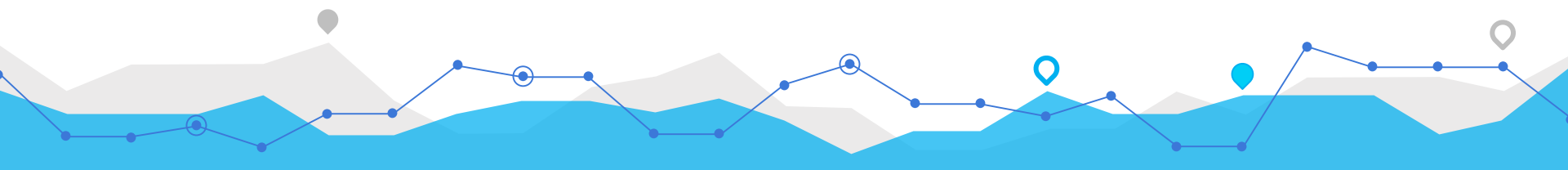




Visallo调研

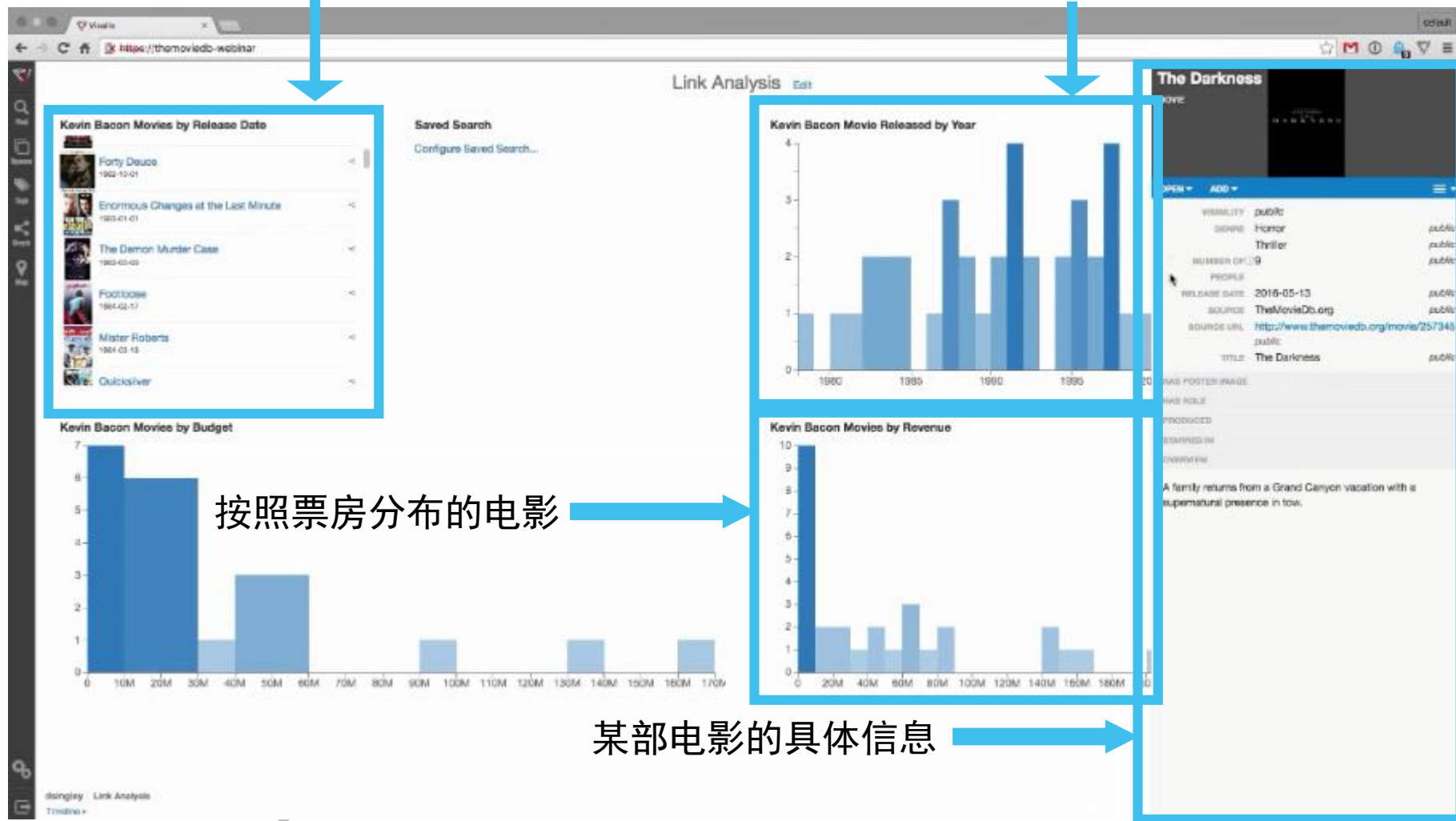
Visallo is an **open source** big data **platform** for analytical tools and visualizations to help people solve **ambiguous problems**.

- 公司：v5 analytics，2014年12月1日成立
- investigative case management platform（调查案例管理平台）
- UI for curating and exploring Big Data(一个大数据策划和探索的用户界面)
- 特点：bring different types of data various data sources throughout your enterprise into one system（处理多源异构数据）
- Secure and collaborative environment for
 - discovering new relationships发现新的关系
 - linking data链接数据
 - viewing data查看数据
- 横向可拓展性
- 细粒度的安全性



演员表

某个演员不同年份发布的电影（可交互）



按照票房分布的电影

某部电影的具体信息



为一个人添加相关的电影

为某部电影添加出演这部电影的演员



Extension Points

后端 (Java)

- Ingestion
- Complete Dataset Analytics
- External Resource Worker
- Graph Property Worker
- Long Running Processes
- Web Plugins



Extension Points

后端:Ingestion

■ 方法一：导入RDF

- 两种不同格式的PDF
 - ✓ NT Format(N-Triples)
 - ✓ RDF XML Import
- 如何导入PDF文件
 - ✓ 后端：`org.visallo.tools.RdfImport`
 - ✓ 前端：用文件上传的方式，自动导入visallo的公共数据集



Extension Points

后端:Ingestion

■ 方法二：code generator

`org.visallo.tools.ontology.ingest.codegen.ModelCodeGen`：

- 连接一个运行中的Visallo实例并且从中获取ontology
 - 利用一个JSON格式的ontology来运行code generator
-
- 利用生成的代码，可以做这些事：
 - 创建实体
 - 创建关系
 - 可视化实体之间的关系



Extension Points

后端:Ingestion

- 方法三：REST (Representational State Transfer)
 - 建立连接：通过发送请求，返回Token的方法来建立连接
 - 创建节点：利用返回的Token，创建节点，大概包括这几个方面：
 - CSRFToken
 - vertexId (新顶点的ID)
 - conceptType (新顶点的概念类型)
 - visibilitySource
 - properties
 - publish
 - 创建节点属性：通过JSON的方式



Extension Points

后端：其他

Graph Property Worker

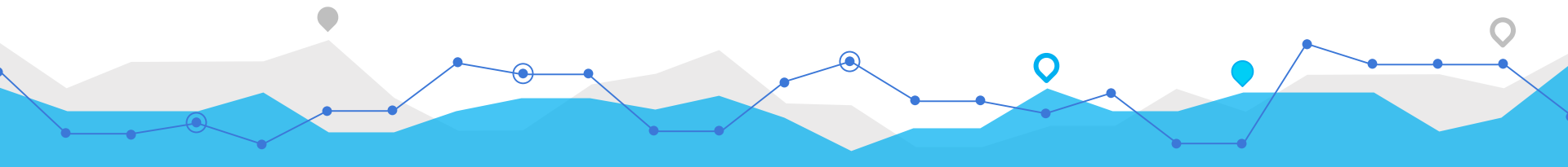
- data enhancement and individual scoring analytics
- 元素/属性的监听器
- 两个方法：
 - isHandled
 - execute

Long Running Processes

- 支持在后台运行程序
- 工作模式类似于Graph Property Workers，从队列中接收一条消息，isHandled决定是否处理它，如果能处理的话，调用processInternal来处理

External Resource Workers

- 有时候需要异步导入数据

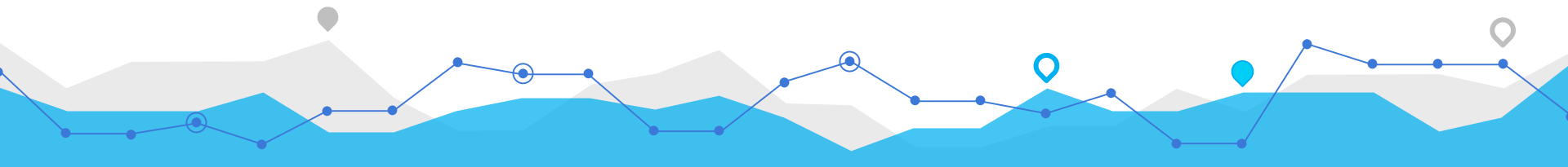


Extension Points

后端

Complete Data Analytics

- 未开源，只能在企业版里获取到
- 功能包括：
 - Classification
 - Clustering
 - Recommendation
 - Statistical Algorithms



Extension Points

前端

- 可以引入自定义的外部依赖：

```
app.registerJavaScript("plugin.js");  
app.registerCss( "plugin.css" );
```

- 扩展点（Extension Points）：用来定义额外的功能，字符串到JavaScript对象的一个映射关系

```
require(['configuration/plugins/registry'], function(registry){  
    registry.registerExtension([extension point name],  
        [extension point object])  
})
```



Extension Points

前端

■ Plugins

- ✓ **Activity**
- ✓ Admin
- ✓ Authentication
- ✓ **Dashboard** Item
- ✓ Dashboard Layout
- ✓ Dashboard Report
- ✓ Dashboard Toolbar
- ✓ Detail Text
- ✓ Detail Toolbar
- ✓ Graph Decorations
- ✓ Graph Edge Class
- ✓ Graph Edge Transformer
- ✓ Graph Export
- ✓ Graph Layout
- ✓ Graph Node Class
- ✓ Graph Node Transformer
- ✓ Graph Options
- ✓ Graph Selector
- ✓ Graph Style
- ✓ Graph View
- ✓ Websocket
- ✓ Layout Component
- ✓ Layout Type
- ✓ Logout
- ✓ Map Options
- ✓ Menubar
- ✓ Search Filters
- ✓ Search Toolbar
- ✓ User Account
- ✓ Vertex Menu
- ✓ Visibility